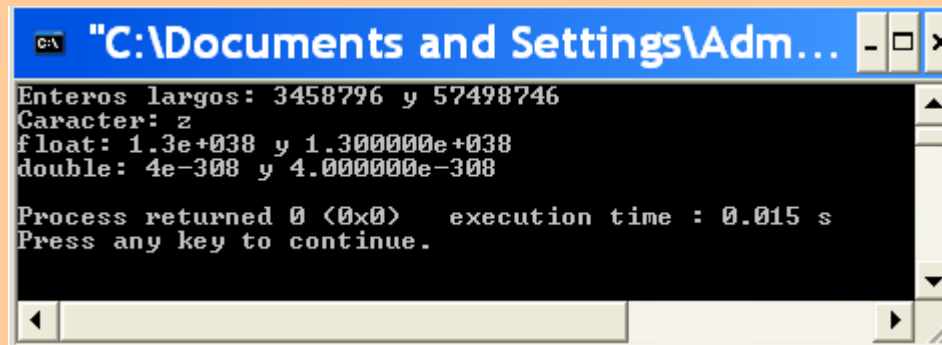


CODIFICACION - PARTE II

```
#include<stdio.h>
main() {
    /* declaración de variables */
    long int i, j;
    char letra;
    float x, y;
    double u, v;
    /* inicialización de
    variables*/
    i = 3458796;
    j = 57498746;
    letra = 'z';
    x = 1.3e38;
    y = 1.3e38;
    u = 4e-308;
    v = 4e-308;
    printf("Enteros largos: %ld y %ld \n", i, j);
    printf("Caracter: %c \n", letra);
    printf("float: %g y %e \n", x, y);
    printf("double: %g y %le \n", u, v); }
```



```
C:\ "C:\Documents and Settings\Adm...
Enteros largos: 3458796 y 57498746
Caracter: z
float: 1.3e+038 y 1.3000000e+038
double: 4e-308 y 4.0000000e-308
Process returned 0 (0x0) execution time : 0.015 s
Press any key to continue.
```

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
/* declaración e  
inicialización de  
variables*/
```

```
float a = 45.375;
```

```
double pi = 3.141592653589793115997963;
```

```
/* salida de a*/
```

```
printf("a: %f \n", a);
```

```
printf("a con 2 decimales: %.2f \n", a);
```

```
printf("a con 7 decimales: %.7f \n", a);
```

```
/*salida de pi*/
```

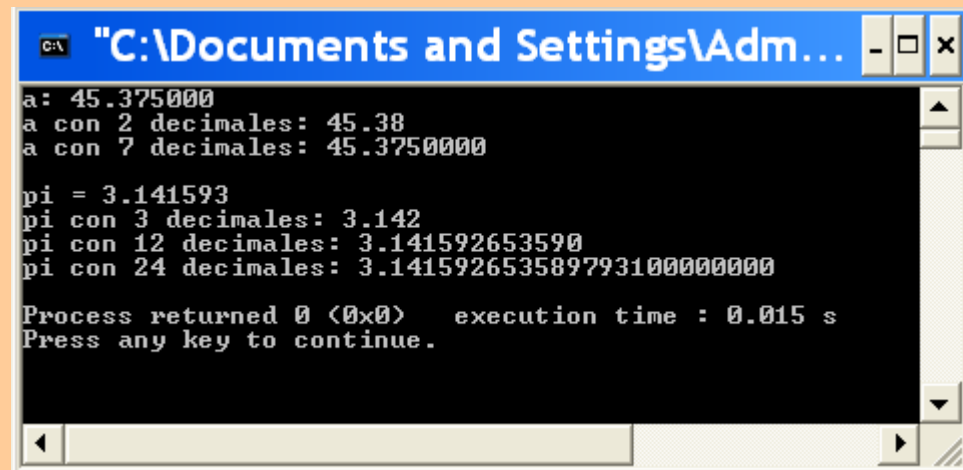
```
printf("\npi = %lf \n", pi);
```

```
printf("pi con 3 decimales: %.3lf \n", pi);
```

```
printf("pi con 12 decimales: %.12lf \n", pi);
```

```
printf("pi con 24 decimales: %.24lf \n", pi);
```

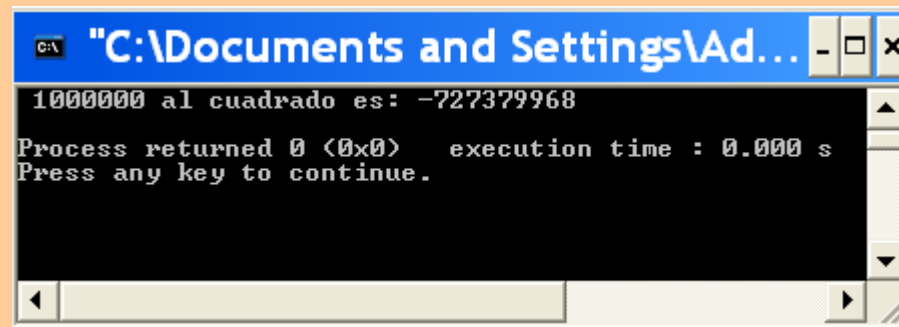
```
}
```



```
C:\Documents and Settings\Adm...  
a: 45.375000  
a con 2 decimales: 45.38  
a con 7 decimales: 45.3750000  
  
pi = 3.141593  
pi con 3 decimales: 3.142  
pi con 12 decimales: 3.141592653590  
pi con 24 decimales: 3.14159265358979310000000000  
  
Process returned 0 (0x0)   execution time : 0.015 s  
Press any key to continue.
```

Error!!!!

```
#include<stdio.h>
main(){
/* Inicializa una variable entera*/
int millon = 1000000;
/* La siguiente linea genera un error,
la variable billon = 10^12 supera a 2^31 */
int billon = millon*millon;
/* imprime dato errado*/
printf(" %d al cuadrado es: %d \n", millon, billon);
}
```



```
C:\ "C:\Documents and Settings\Ad... - □ ×
1000000 al cuadrado es: -727379968
Process returned 0 (0x0) execution time : 0.000 s
Press any key to continue.
```

Lectura de datos

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
/* Declaración de  
variables*/
```

```
int antigüedad;
```

```
float sueldo;
```

```
/* lee antigüedad*/
```

```
printf("Ingrese su antigüedad en meses: ");
```

```
scanf("%d", &antigüedad);
```

```
/* lee sueldo*/
```

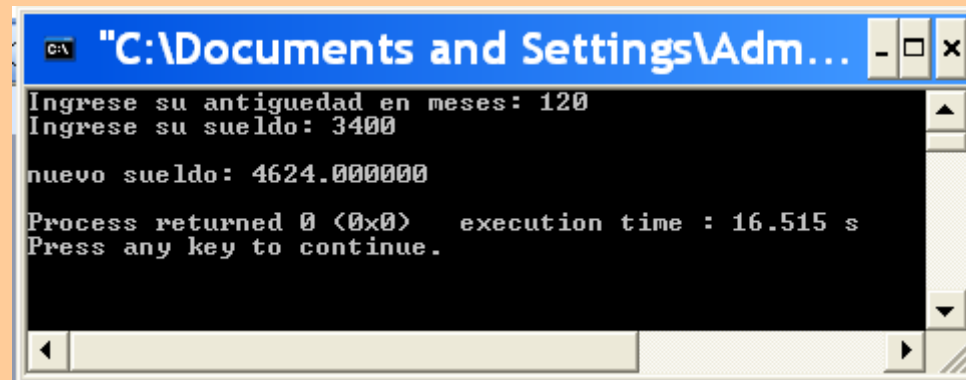
```
printf("Ingrese su sueldo: ");
```

```
scanf("%f", &sueldo);
```

```
/* Imprime nuevo sueldo*/
```

```
printf("\nnuevo sueldo: %f \n", sueldo*(1+0.3*antigüedad/100));
```

```
}
```

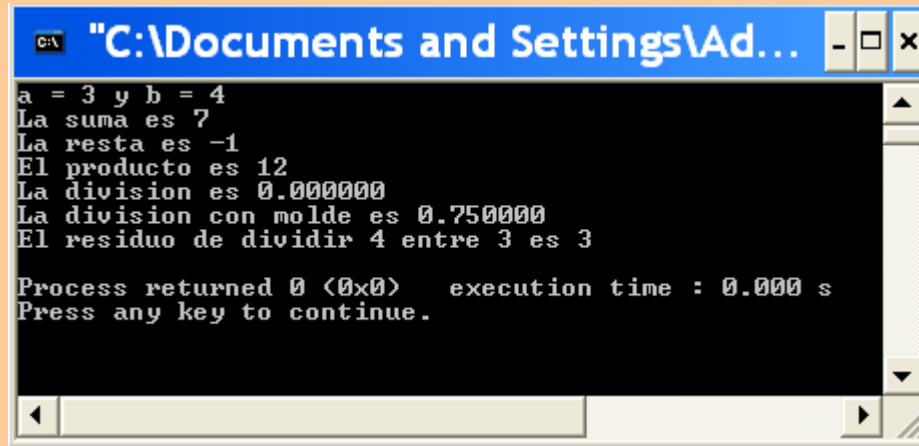


```
C:\ "C:\Documents and Settings\Adm...  
Ingrese su antigüedad en meses: 120  
Ingrese su sueldo: 3400  
  
nuevo sueldo: 4624.000000  
  
Process returned 0 (0x0)   execution time : 16.515 s  
Press any key to continue.
```

```

#include<stdio.h>
main()
{
/* Declaraci_ón de
variables*/
int a = 3;
int b = 4;
int suma = a+b;
int resta = a-b;
int producto = a*b;
double division = a/b;
int residuo = a %b;
/* imprime contenido de variables*/
printf("a = %d y b = %d \n", a, b);
printf("La suma es %d \n", suma);
printf("La resta es %d \n", resta);
printf("El producto es %d \n", producto);
printf("La division es %lf \n", division);
printf("La division con molde es %lf \n", (double)a/b);
printf("El residuo de dividir %d entre %d es %d \n", b, a, residuo);}

```



```

C:\Documents and Settings\Ad...
a = 3 y b = 4
La suma es 7
La resta es -1
El producto es 12
La division es 0.000000
La division con molde es 0.750000
El residuo de dividir 4 entre 3 es 3

Process returned 0 (0x0)   execution time : 0.000 s
Press any key to continue.

```

Operadores Relacionales

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
/* Declaracion de  
variables*/
```

```
int a, b, c;
```

```
/* Inicializacion de  
variables*/
```

```
a = 2;
```

```
b = 5;
```

```
c = -3;
```

```
printf("a = %d, b = %d y c = %d \n\n", a, b, c);
```

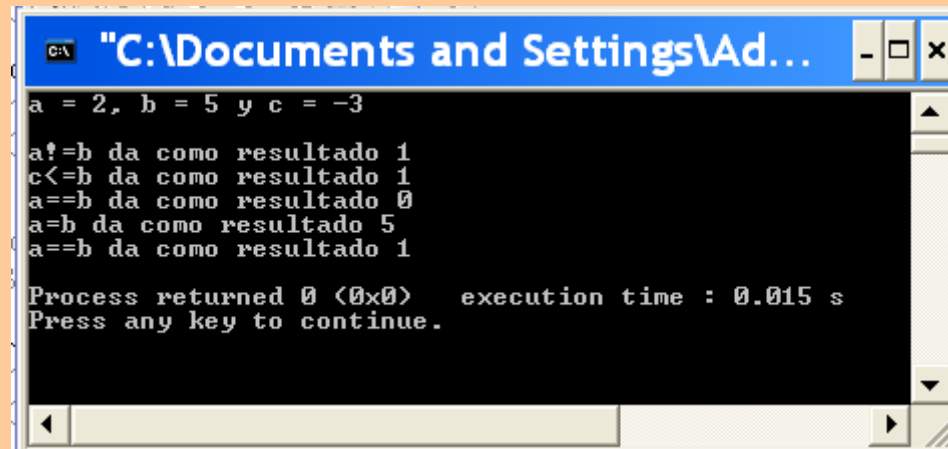
```
printf("a!=b da como resultado %d \n", a!=b);
```

```
printf("c<=b da como resultado %d \n", c<=b);
```

```
printf("a==b da como resultado %d \n", a==b);
```

```
printf("a=b da como resultado %d \n", a=b);
```

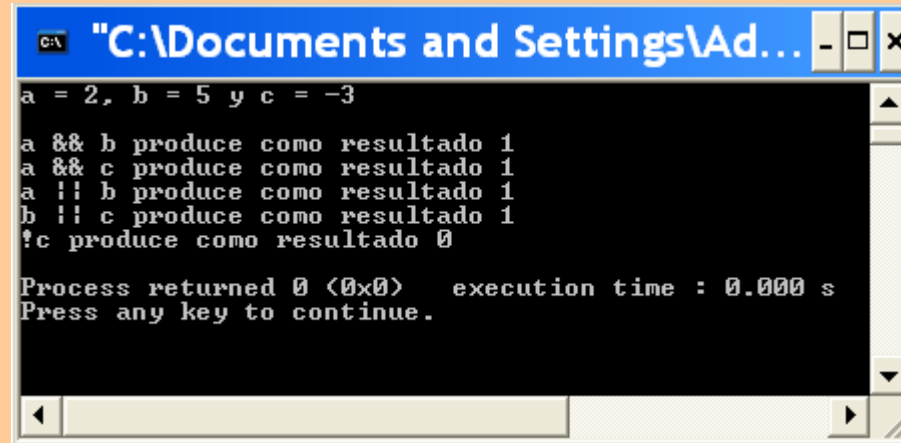
```
printf("a==b da como resultado %d \n", a==b);} 
```



```
C:\Documents and Settings\Ad...
a = 2, b = 5 y c = -3
a!=b da como resultado 1
c<=b da como resultado 1
a==b da como resultado 0
a=b da como resultado 5
a==b da como resultado 1
Process returned 0 (0x0)   execution time : 0.015 s
Press any key to continue.
```

Operadores Lógicos

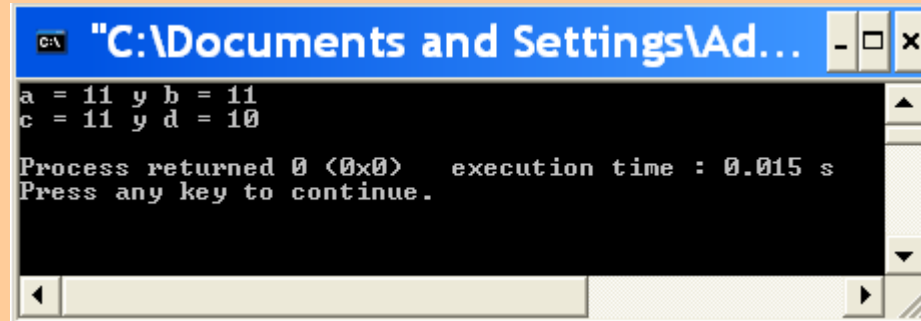
```
main()
{
    /* Declaracion de
    variables*/
    int a, b, c;
    /* Inicializacion de
    variables*/
    a = 2;
    b = 5;
    c = -3;
    printf("a = %d, b = %d y c = %d \n\n", a, b, c);
    printf("a && b produce como resultado %d \n", a && b);
    printf("a && c produce como resultado %d \n", a && c);
    printf("a || b produce como resultado %d \n", a || b);
    printf("b || c produce como resultado %d \n", b || c);
    printf("!c produce como resultado %d \n", !c);
}
```



```
C:\Documents and Settings\Ad...
a = 2, b = 5 y c = -3
a && b produce como resultado 1
a && c produce como resultado 1
a || b produce como resultado 1
b || c produce como resultado 1
!c produce como resultado 0
Process returned 0 (0x0)   execution time : 0.000 s
Press any key to continue.
```

Post y Preincremento

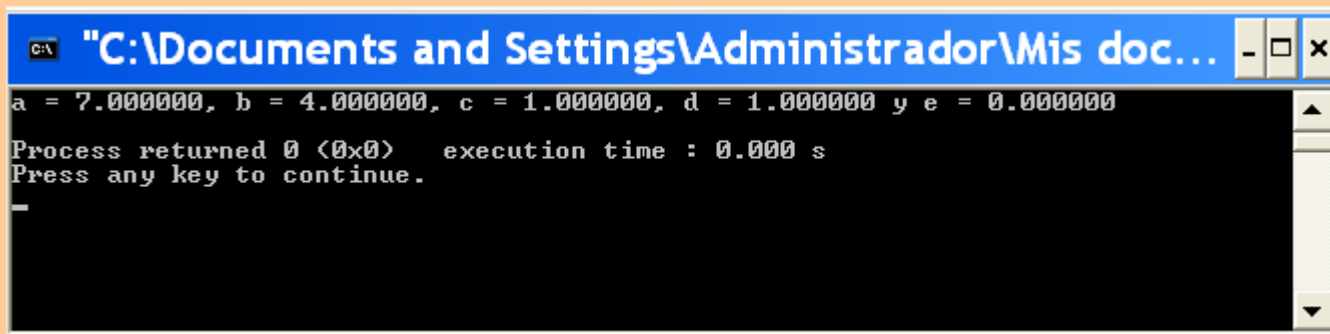
```
#include<stdio.h>
main()
{
/* Declaracion de
variables*/
int a, b, c, d;
/* Inicializacion de variables*/
a = 10;
c = 10;
/*Operador de preincremento */
b = ++a;
printf("a = %d y b = %d \n", a, b);
/*Operador de postincremento */
d = c++;
printf("c = %d y d = %d \n", c, d);
}
```



```
C:\Documents and Settings\Ad...
a = 11 y b = 11
c = 11 y d = 10
Process returned 0 (0x0)   execution time : 0.015 s
Press any key to continue.
```


Precedencia de Operadores

```
#include<stdio.h>
main()
{
/* Declaracion de variables*/
float a, b, c, d, e;
/* Inicializacion de variables*/
a = 1.0 + 2.0 * 3.0;
b = 6.0 / 3.0 * 2.0;
c = 6.0 / ( 3.0 * 2.0 );
d = ! 1 < 4 + 8;
e = ! ( 1 < 4 + 8 );
printf("a = %f, b = %f, c = %f, d = %f y e = %f\n", a, b, c , d, e);
}
```



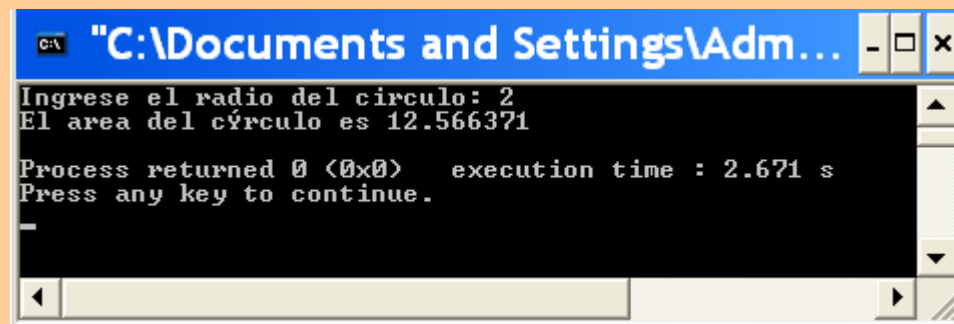
```
C:\ "C:\Documents and Settings\Administrador\Mis doc... - □ x
a = 7.000000, b = 4.000000, c = 1.000000, d = 1.000000 y e = 0.000000
Process returned 0 (0x0)   execution time : 0.000 s
Press any key to continue.
_
```

Funciones Matemáticas

| Función | Descripción | Argumentos | Resultado |
|-----------------------|--------------------------|----------------|-----------|
| <code>sqrt(x)</code> | raíz cuadrada \sqrt{x} | double | double |
| <code>pow(x,y)</code> | potencia x^y | double, double | double |
| <code>exp(x)</code> | exponencial | double | double |
| <code>log(x)</code> | logaritmo natural | double | double |
| <code>sin(x)</code> | seno | double | double |
| <code>cos(x)</code> | coseno | double | double |
| <code>tan(x)</code> | tangente | double | double |
| <code>asin(x)</code> | arco seno | double | double |
| <code>acos(x)</code> | arco coseno | double | double |
| <code>atan(x)</code> | arco tangente | double | double |
| <code>abs(x)</code> | valor absoluto de entero | int | int |
| <code>fabs(x)</code> | valor absoluto de double | double | double |
| <code>ceil(x)</code> | redondea hacia arriba x | double | int |
| <code>floor(x)</code> | redondea hacia abajo x | double | int |

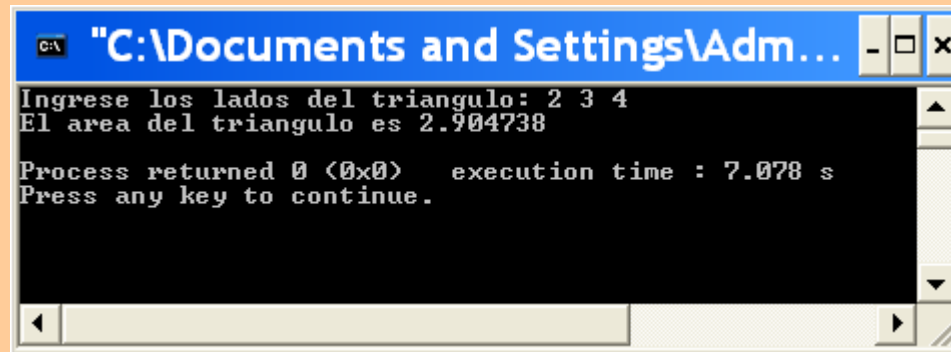
Area del círculo

```
#include<stdio.h>
#include<math.h>
main()
{
/* Declaracion de variables*/
double r, area;
printf("Ingrese el radio del círculo: ");
scanf("%lf", &r);
area =M_PI*pow(r,2); /*M_PI esta definido en math.h*/
printf("El área del círculo es %lf \n", area);
}
```



Area del triángulo

```
#include<stdio.h>
#include<math.h>
main()
{
    /* Declaración de variables*/
    double a, b, c, s, area;
    printf("Ingrese los lados del triangulo: ");
    scanf("%lf %lf %lf", &a, &b, &c);
    /* Calcula el area con la formula de Heron*/
    s = (a+b+c)/2; /* semiperimetro*/
    area = sqrt(s*(s-a)*(s-b)*(s-c));
    printf("El area del triangulo es %lf \n", area);
}
```



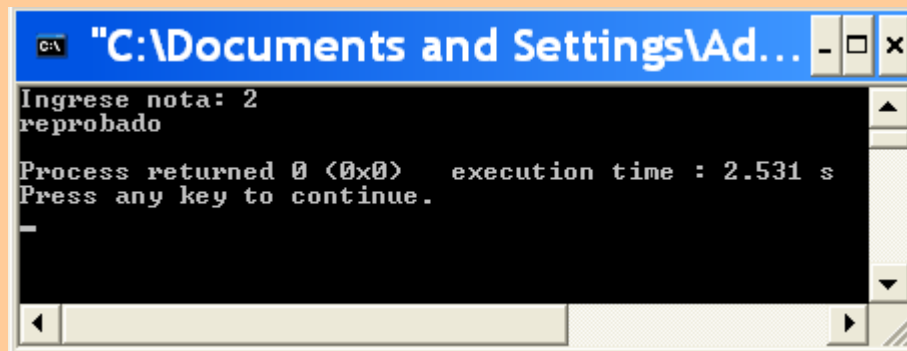
The screenshot shows a Windows command prompt window with the title bar "C:\Documents and Settings\Adm...". The window contains the following text:

```
C:\>
Ingrese los lados del triangulo: 2 3 4
El area del triangulo es 2.904738

Process returned 0 (0x0)   execution time : 7.078 s
Press any key to continue.
```

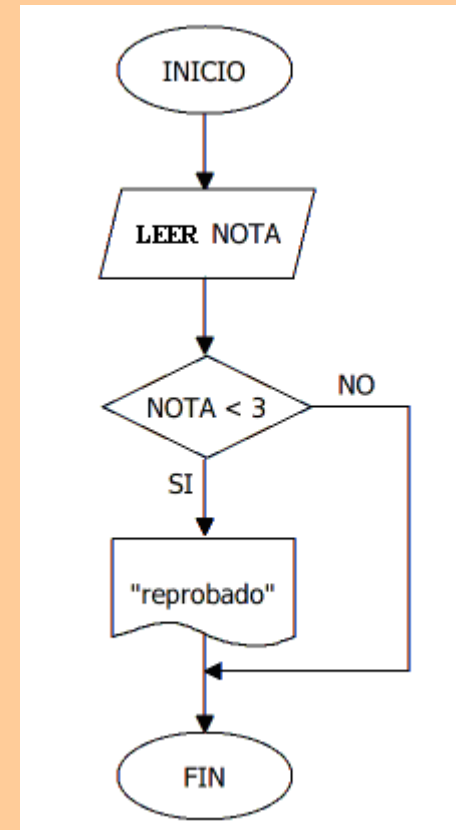
Estructura selectiva simple

```
#include<stdio.h>
main()
{
/*Declaracion de variables*/
float NOTA;
/* leer nota de alumno*/
printf("Ingrese nota: ");
scanf("%f", &NOTA);
/*condicional */
if (NOTA < 3.0)
    printf("reprobado \n");
}
```



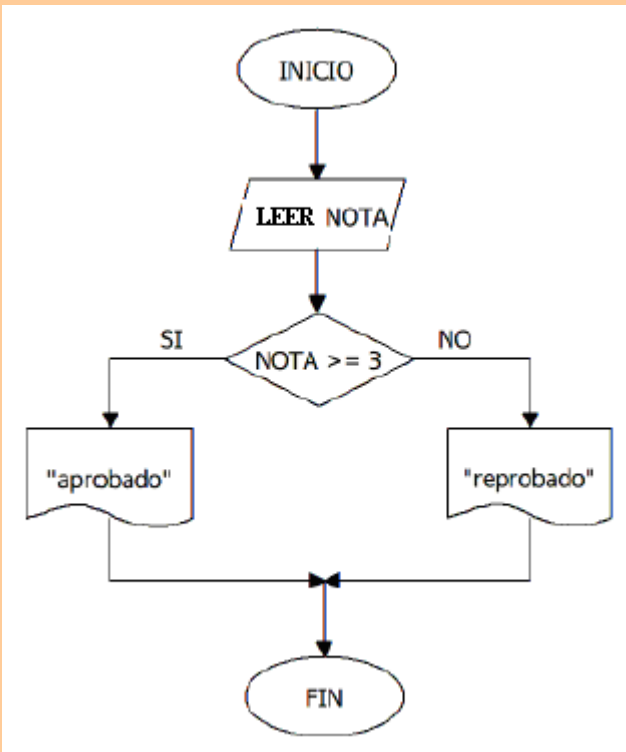
```
C:\Documents and Settings\Ad...
Ingrese nota: 2
reprobado

Process returned 0 (0x0)   execution time : 2.531 s
Press any key to continue.
```



Estructura selectiva doble

```
#include<stdio.h>
main()
{ /*Declaracion de variables*/
float NOTA;
/* Ingresa nota*/
printf("Ingresa nota: ");
scanf("%f", &NOTA);
/* condicional*/
if (NOTA >= 3.0)
printf("aprobado \n");
else
printf("reprobado \n");
}
```

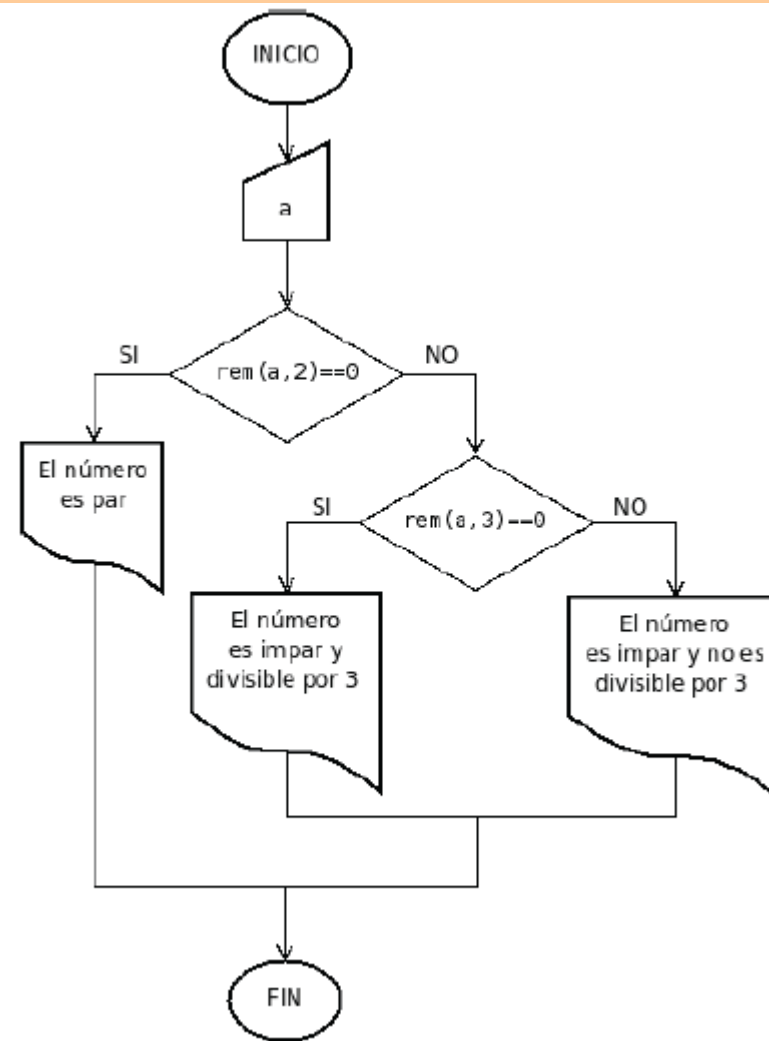


The screenshot shows a Windows command prompt window with the title bar "C:\Documents and Settings\Adm...". The window contains the following text: "Ingresa nota: 5", "aprobado", "Process returned 0 (0x0) execution time : 3.296 s", and "Press any key to continue.". The user has entered the number 5, and the program has output "aprobado".

Estructura selectiva anidada

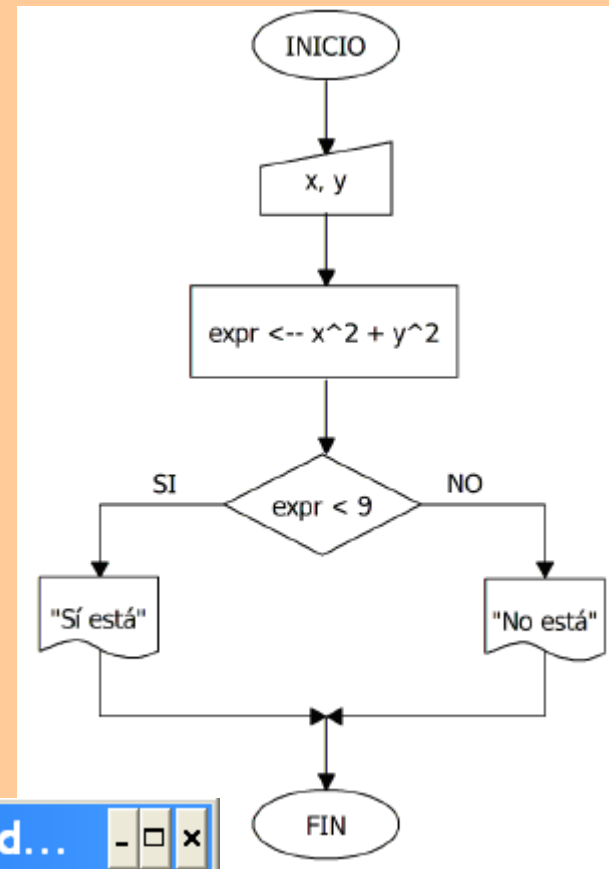
Dado un entero, determinar si es par y en caso de no serlo, ver si es divisible por 3.

```
#include<stdio.h>
main(){
/*Declaracion de variables*/
int a;
/* leer entero*/
printf("Ingrese entero: ");
scanf("%d", &a);
/* condicional if-else*/
if (a %2==0)
{
    printf(" %d es par", a);
}
else
{if (a %3==0)
{printf(" %d es divisible por tres", a);}
else
{printf(" %d no es divisible por tres",a);}
}
}
```



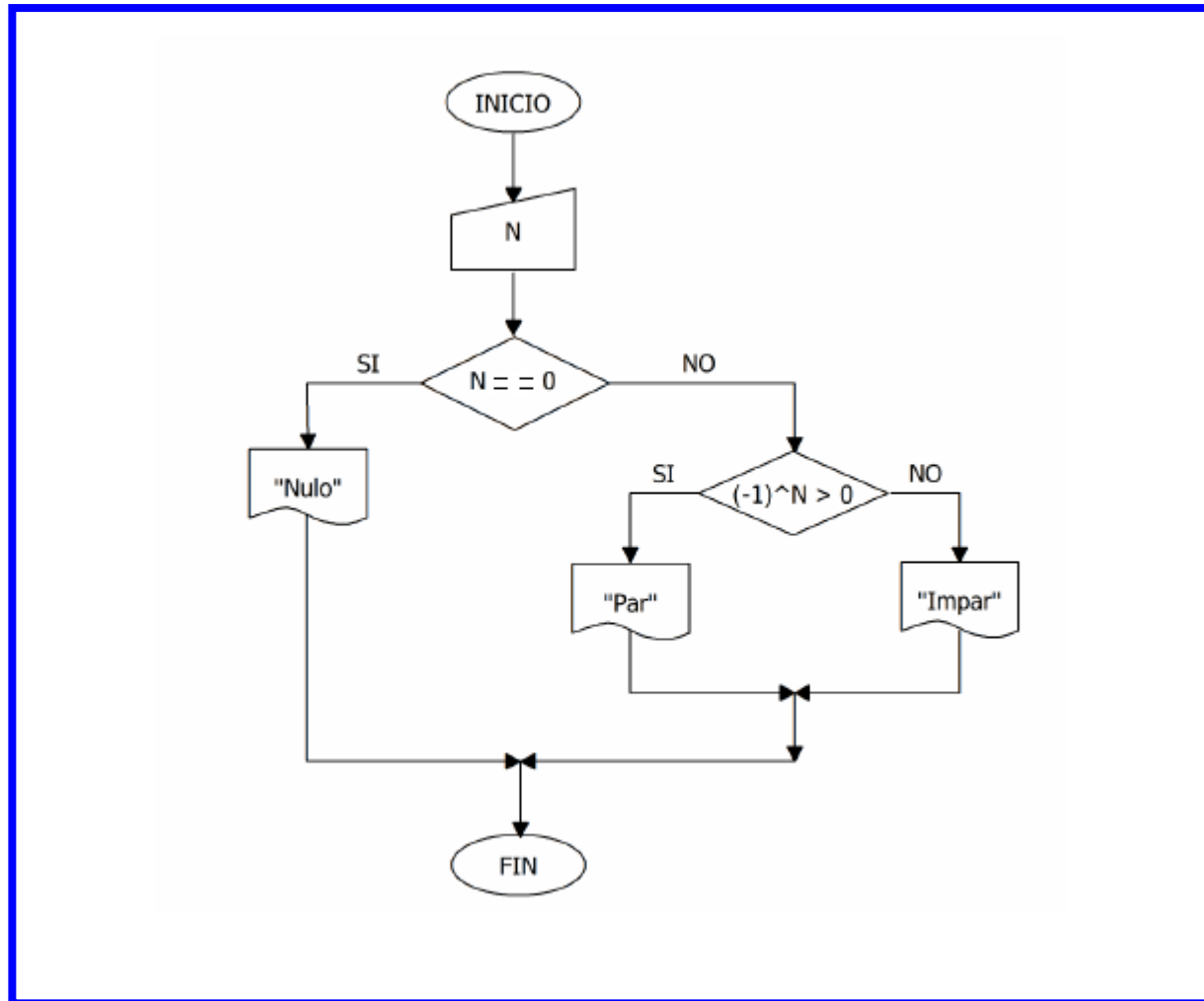
Determinar si el punto de coordenadas (x,y) está en el interior de la circunferencia con centro en el origen y radio 3.

```
#include<stdio.h>
#include <math.h>
main()
{ /*Declaracion de variables*/
float x,y;
float expr;
/* Ingresar coordenadas*/
printf("Ingrese x e y: ");
scanf("%f", &x);
scanf("%f", &y);
expr=pow(x,2)+pow(y,2);
if(expr<9)
printf("Si esta");
else
printf("No esta");
}
```



The screenshot shows a Windows command prompt window with the title bar "C:\Documents and Settings\Ad...". The window contains the following text: "Ingrese x e y: 1.5 4", "No esta", "Process returned 0 (0x0) execution time : 6.765 s", and "Press any key to continue.". The cursor is positioned at the bottom of the window, ready for input.

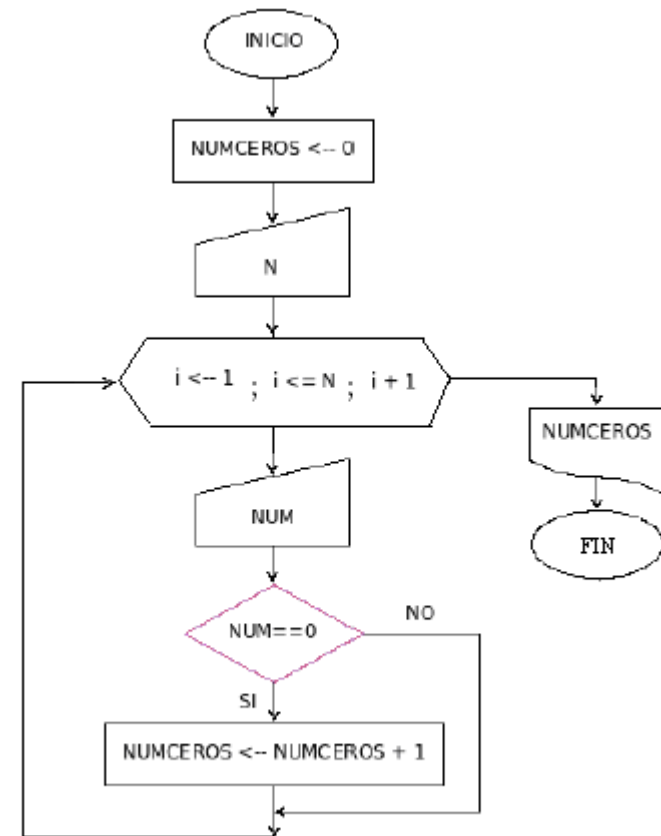
Dado un número entero imprimir si es par, impar o nulo



Estructura de control repetitiva - FOR

Realice un programa en C que lea N números enteros desde el teclado y cuente cuántos de ellos son ceros.

```
#include<stdio.h>
main()
{
    int k, N, NUM, NUMCEROS;
    NUMCEROS = 0; /* variable contadora*/
    printf("Numero de datos a ingresar: ");
    scanf("%d", &N);
    for (k=1; k<=N; k++) {
        printf("Ingrese numero: ");
        scanf("%d", &NUM);
        if (NUM==0)
            NUMCEROS = NUMCEROS + 1;
    }
    printf("Numero de ceros: %d \n", NUMCEROS);
}
```



The screenshot shows a Windows command prompt window with the title bar "C:\Documents and Settings\Adm...". The window contains the following text:

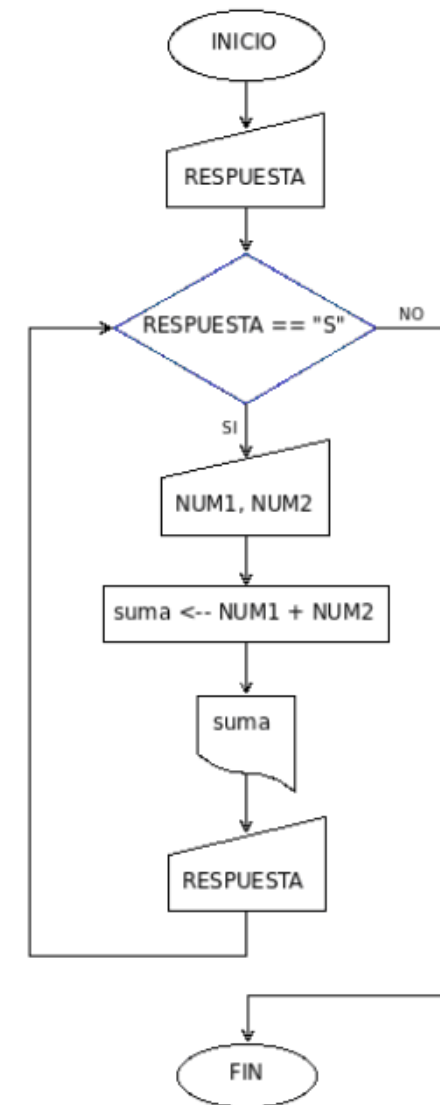
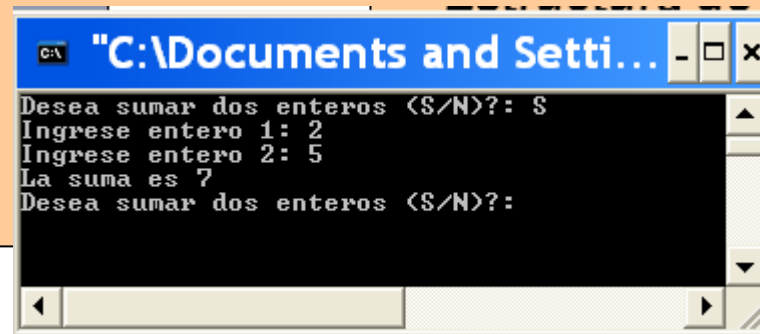
```
Numero de datos a ingresar: 3
Ingrese numero: 1
Ingrese numero: 0
Ingrese numero: 4
Numero de ceros: 1

Process returned 0 (0x0)   execution time : 5.890 s
Press any key to continue.
```

Estructura de control repetitiva - WHILE

Mostrar en pantalla la suma de dos números enteros leídos por teclado. Repetir el proceso cuantas veces se desee.

```
#include<stdio.h>
main(){
int NUM1, NUM2, suma;
char RESPUESTA;
printf("Desea sumar dos enteros (S/N)? : ");
scanf(" %c", &RESPUESTA);
while (RESPUESTA=='S') {
    printf("Ingrese entero 1: ");
    scanf(" %d", &NUM1);
    printf("Ingrese entero 2: ");
    scanf(" %d", &NUM2);
    suma = NUM1 + NUM2;
    printf("La suma es %d \n", suma);
    printf("Desea sumar dos enteros (S/N)? : ");
    scanf(" %c", &RESPUESTA);
}
```



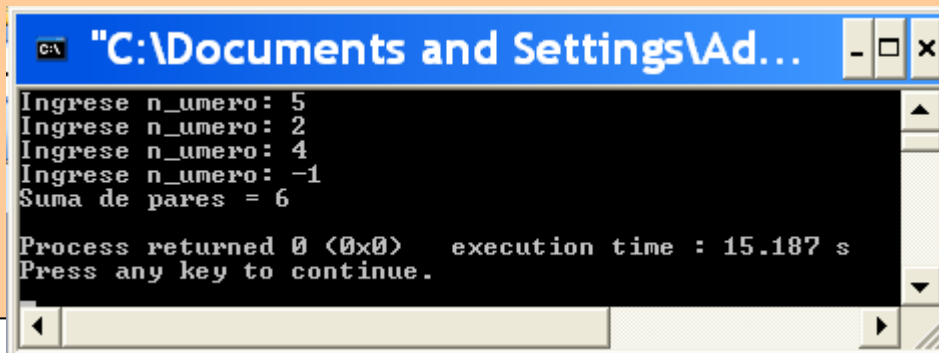
Calcular el promedio de los primeros N números naturales, usando estructuras repetitivas WHILE y FOR.

```
#include<stdio.h>
main()
{
    int N, NUM = 1, SUM = 0;
    float PROM;
    printf("Ingrese N: ");
    scanf(" %d", &N);
    while (NUM<=N)
    {
        SUM = SUM + NUM;
        NUM = NUM + 1;
    }
    PROM = (float)SUM/(float)N;
    printf("Promedio = %g \n", PROM);
}
```

```
#include<stdio.h>
main()
{
    int N, i, SUM = 0;
    float PROM;
    printf("Ingrese N: ");
    scanf(" %d", &N);
    for (i=1;i<=N;i++)
    {
        SUM = SUM + i;
    }
    PROM = (float)SUM/(float)N;
    printf("Promedio = %g \n", PROM);
}
```

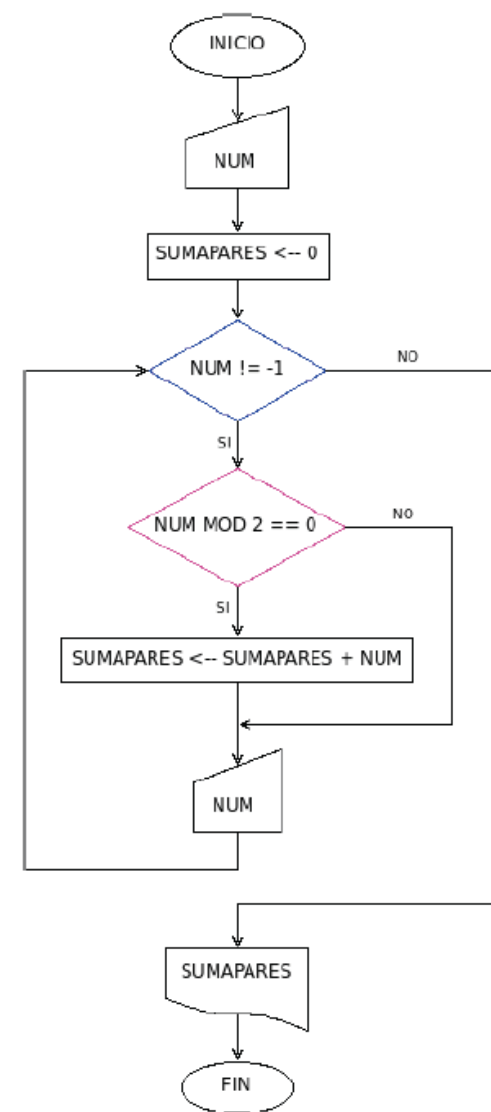
Programa que reciba por teclado enteros positivos y sume los Pares, hasta el ingreso del -1 (centinela).

```
#include<stdio.h>
main(){
int NUM, SUMAPARES = 0;
float PROM;
printf("Ingrese n_umero: ");
scanf(" %d", &NUM);
while (NUM!=-1)
{
if (NUM %2==0)
SUMAPARES = SUMAPARES + NUM;
printf("Ingrese n_umero: ");
scanf(" %d", &NUM);
}
printf("Suma de pares = %d \n", SUMAPARES);}
```



```
C:\Documents and Settings\Ad...
Ingrese n_umero: 5
Ingrese n_umero: 2
Ingrese n_umero: 4
Ingrese n_umero: -1
Suma de pares = 6

Process returned 0 (0x0)   execution time : 15.187 s
Press any key to continue.
```



Generar los primeros N términos de la sucesión
17 ; 15 ; 18 ; 16 ; 19 ; 17 ; 20 ; 18 ; 21 ;

```
#include<stdio.h>
```

```
main(){
```

```
int NT, TER = 17, CONT = 1, BAN = 1;
```

```
printf("Ingrese numero de terminos: ");
```

```
scanf(" %d", &NT);
```

```
while (CONT<=NT) {
```

```
printf(" %d",TER);
```

```
if (BAN==1){
```

```
    TER = TER - 2;
```

```
    BAN = 2;
```

```
}
```

```
else {
```

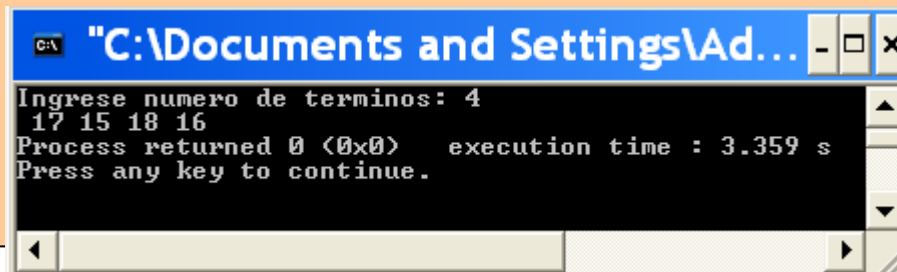
```
    TER = TER + 3;
```

```
    BAN = 1;
```

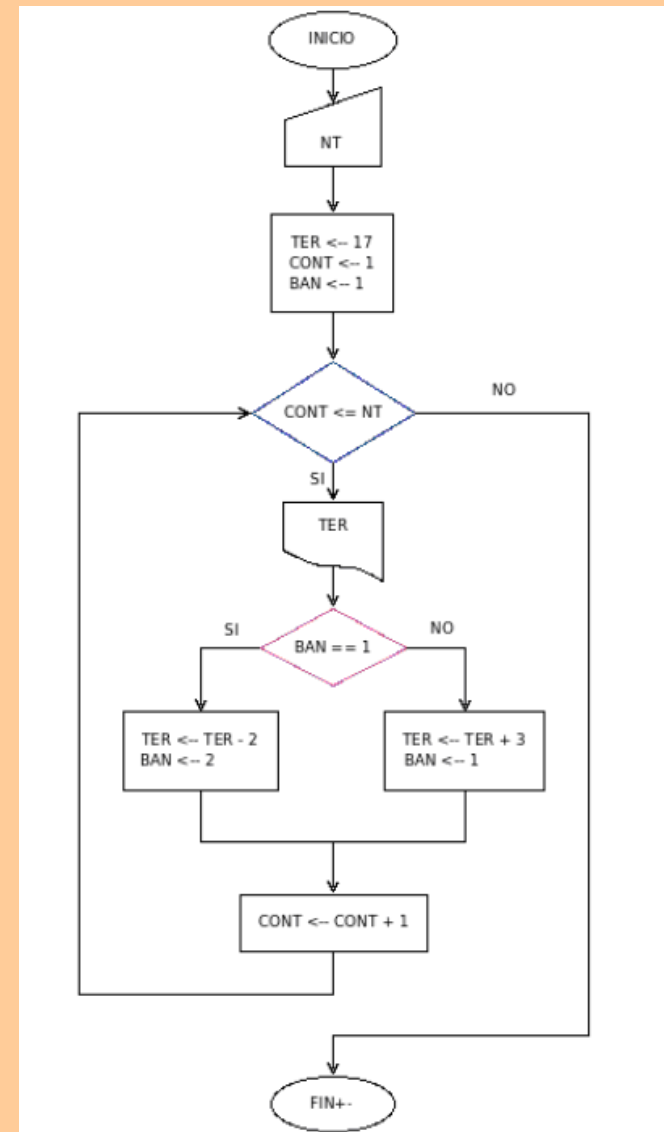
```
}
```

```
CONT = CONT + 1;}
```

```
}
```



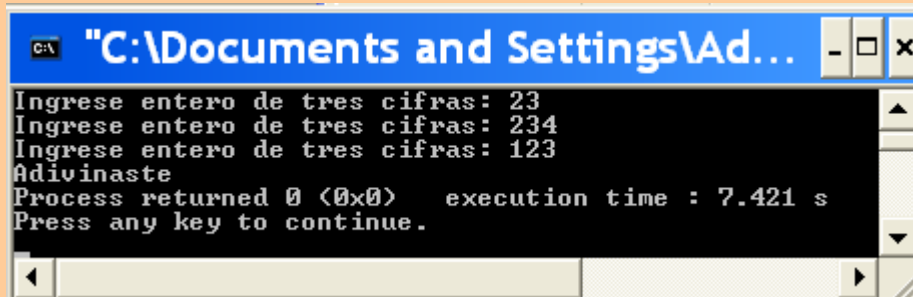
```
C:\Documents and Settings\Ad...
Ingrese numero de terminos: 4
17 15 18 16
Process returned 0 (0x0) execution time : 3.359 s
Press any key to continue.
```



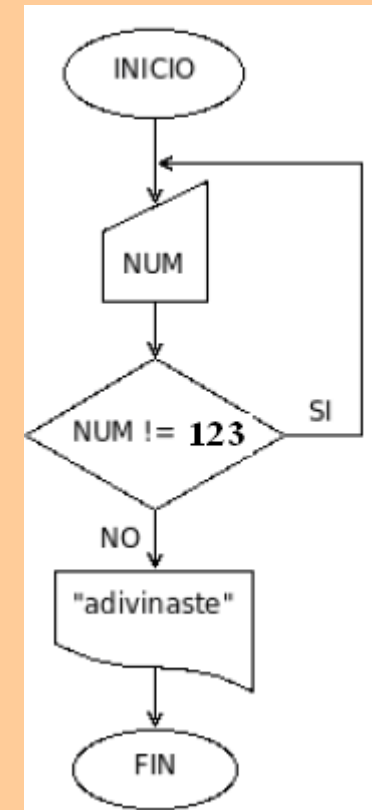
Estructura de control repetitiva – DO WHILE

Solicite al usuario un entero; cuando coincida con el número 123, imprimir "adivinaste".

```
#include<stdio.h>
main()
{
    int NUM;
    do
    {
        printf("Ingrese entero de tres cifras: ");
        scanf(" %d", &NUM);
    } while (NUM != 123);
    printf("Adivinaste");
}
```



```
C:\Documents and Settings\Ad...
Ingrese entero de tres cifras: 23
Ingrese entero de tres cifras: 234
Ingrese entero de tres cifras: 123
Adivinaste
Process returned 0 (0x0)   execution time : 7.421 s
Press any key to continue.
```



Calcular el epsilon de la máquina (IEEE-754)

En aritmética de punto flotante el epsilon de la máquina ε se define como el menor valor almacenado en la máquina que satisface $1 + \varepsilon > 1$; es decir, el menor número que el computador reconoce como mayor a cero.

$$\begin{array}{lcl} 1 = \underbrace{0\textcolor{red}{0}1111111}_{8 \text{ bits}} \underbrace{000000000000000000000000}_{23 \text{ bits}} & & \\ 1 + \varepsilon = \underbrace{0\textcolor{red}{0}1111111}_{8 \text{ bits}} \underbrace{000000000000000000000000\textcolor{red}{1}}_{23 \text{ bits}} & \Rightarrow & \varepsilon = 2^{-23} \end{array}$$

$$1.0 + \frac{1}{2} = 1.5$$

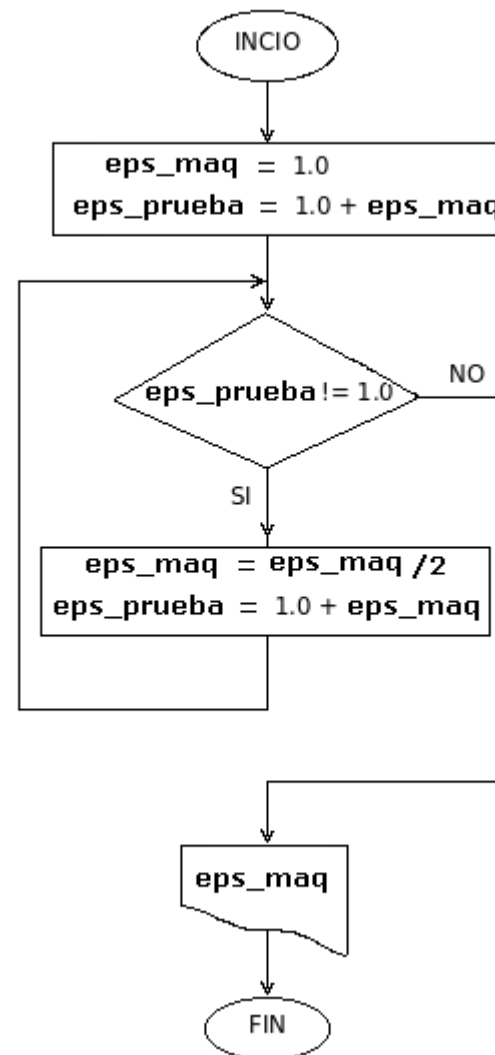
$$1.0 + \frac{1}{2^2} = 1.25$$

$$1.0 + \frac{1}{2^3} = 1.125$$

$$1.0 + \frac{1}{2^4} = 1.0625$$

⋮

$$1.0 + \frac{1}{2^n} = 1.0$$



Dado un valor de x, calcular el valor de la función:

$$f(x) = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \frac{x^5}{5!} + \dots$$

Considerar los términos de la serie que en valor absoluto son mayores o iguales a 10^{-3} .

Análisis de las Variables:

i: variable de tipo entero, genera el exponente y el factorial

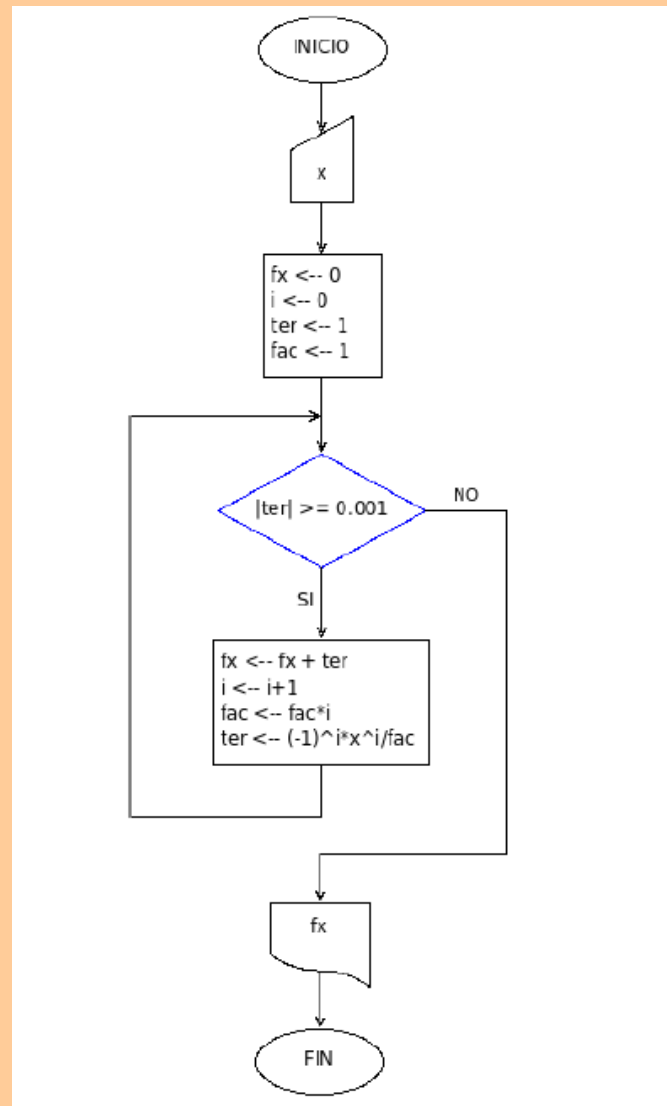
ter: variable de tipo real, almacena cada término de la serie

$$\text{ter} = (-1)^i * (x^i / i!)$$

fac: variable de tipo entera, almacena el factorial

fx: variable de tipo real, acumula la suma de los términos de la serie

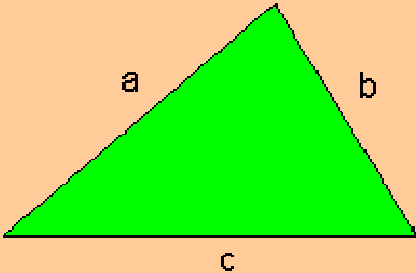
$$\text{fx} = \text{fx} + \text{t}$$



Anexo: Formula de Herón:

Herón de Alejandría vivió hacia el siglo III a. La llamada fórmula de Herón, nos permite calcular el área de un triángulo conocidos los tres lados.

Si llamamos **s** al semiperímetro y **a**, **b**, **c** a los tres lados:



A diagram of a triangle with its interior filled with a bright green color. The three sides are labeled with lowercase letters: 'a' for the left side, 'b' for the right side, and 'c' for the bottom side.

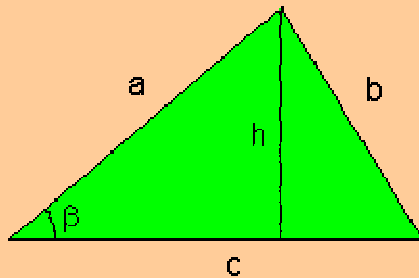
Llamando al semiperímetro

$$s = \frac{a + b + c}{2}$$

entonces el área puede expresarse como

$$A = \sqrt{s(s - a)(s - b)(s - c)}$$

Demostración:



La fórmula clásica para el área del triángulo

nos dice que $A = c \cdot h / 2$;

$$A = c \cdot a \cdot \sin(\beta) / 2.$$

Por el teorema de coseno

$$b^2 = a^2 + c^2 - 2ac \cdot \cos(\beta).$$

Despejando $\cos(\beta)$ de la última ecuación y sustituir $\sin(\beta)$ en la anterior.

Tenemos pues que $\cos(\beta) = (a^2 + c^2 - b^2) / (2ac)$, y como $\sin^2(\beta) = 1 - \cos^2(\beta)$ entonces:

$$\sin(\beta) = \sqrt{1 - \frac{(a^2 + c^2 - b^2)^2}{4a^2c^2}} \quad \text{o} \quad \sin(\beta) = \sqrt{\frac{4a^2c^2 - (a^2 + c^2 - b^2)^2}{4a^2c^2}}$$

Teniendo en cuenta que el numerador es una diferencia de cuadrados y el denominador un cuadrado obtenemos:

$$\begin{aligned}\text{sen}(\beta) &= \text{raíz}[(2ac-(a^2+c^2-b^2))*(2ac+(a^2+c^2-b^2))]/(2ac) \\ &= \text{raíz}[(b^2-(a-c)^2)*((a+c)^2-b^2)]/(2ac)\end{aligned}$$

Sustituyendo ahora en la fórmula del área, tenemos que $A = \text{raíz}[(b^2-(a-c)^2)*((a+c)^2-b^2)]/4$ y utilizando de nuevo la descomposición de la diferencia de cuadrados como suma por diferencia, nos queda:

$$A = \frac{\sqrt{(b+a-c)(b-a+c)(a+c+b)(a+c-b)}}{4}$$

Finalmente, introducimos el 4 dentro de la raíz quedando 16, y si observamos que $(b+a-c)/2 = (s-c)/2$, y que $(b-a+c)/2 = (s-a)/2$ y así sucesivamente, llegamos a la fórmula final:

$$A = \sqrt{s(s-a)(s-b)(s-c)} \quad \text{qed}$$